* **How to Create an Table student write an SQL Query ?**

**CREATE TABLE student(Rollno INT PRIMARY KEY, NAME VARCHAR(30),Branch VARCHAR(40));**

* **How to Create a Exam table with Foreign key on rollno write a SQL Query ?**

**CREATE TABLE Exam (Rollno INT,S\_code VARCHAR(30),**

**Marks INT,**

**P\_code VARCHAR(30),**

**foreign key(Rollno) references student(Rollno));**

* **What is SQL Key Constraints write an Example of SQL Key Constraints ?**

**CREATE TABLE table\_name (column1 data\_type constraint,**

**column2 data\_type constraint,**

**column3 data\_type constraint);**

**Unique key ( column can not have a NULL values)**

**Primary key ( combination of not NULL and unique )**

**Foreign key ( connect two table for use another table record )**

* **What is SQL View Create a View of Student Table ?**

**select \*from exam;**

**How to Create a Table user write a SQL query ?**

**CREATE TABLE USER (Firstname VARCHAR(20),Lastname VARCHAR(20),address VARCHAR(50),city VARCHAR(30),age INT);**

* **What is SQL and How to Create a table with Forign Key ?**

**CREATE TABLE Exam (Rollno INT,S\_code VARCHAR(30),**

**Marks INT,**

**P\_code VARCHAR(30),**

**foreign key(Rollno) references student(Rollno));**

* **What is store Proceedure write a query of creating store Proceedure ?**

**DELIMITER $$**

**CREATE**

**procedure procedure\_name ()**

**BEGIN**

**select \*from table\_name where id=100; -- (ENTER YOUR TABLE ID NUMBER)**

**END $$**

* **What is save Point How to Create a save Point write a Query ?**

**START TRANSACTION;**

**delete from table\_name where id=2 -- (ENTER YOUR TABLE ID)**

**savepoint a;**

**update name='rahul' from table\_name where id = 3;**

**rollback;**

* **What is trigger and how to Create a Trigger in SQL ?**

**create**

**trigger `database\_name`.`update\_name` before delete**

**on `database\_name`.`table\_name`**

**for each row begin**

**insert into new\_table\_name(id,name,email) values (old.id,old.name, old.email);**

**end $$**

* **What is DDL Interpreter ?**

**THE DDL COMMANDS ARE:**

**CREATE DATABASE**

**ALTER TABLE**

**DROP TABLE OR DATABASE**

**TRUNCATE TABLE**

* **Get First\_Name from employee table using alias name “Employee Name”.**

**select first\_name as employee\_name from employee;**

* **Get FIRST\_NAME, Joining year, Joining Month and Joining Date from employee table.**

**select first\_name , year (joining\_year),month(joining\_month),date(joining\_date) from table\_name;**

* **Get all employee details from the employee table order by First\_Name Ascending and Salary descending.**

**select \*from employee order by First\_name;**

**select \*fron employee order by salary desc;**

* **Get employee details from employee table whose first name contains ‘o’.**

**select \*from employee where name like '%o%';**

* **Get employee details from employee table whose joining month is “January”.**

**select \*from employee where month = 'january'(joining\_month);**

* **Get department, total salary with respect to a department from employee table order by total salary descending.**

**select department,sum(salary) total\_salary from employee group by department order by total\_salary desc;**

* **Get department wise maximum salary from employee table order by salary ascending**

**select department ,max(salary) max\_salary from employee group by department order by max\_salary;**

* **Select first\_name, incentive amount from employee and incentives table for those employees who have incentives and incentive amount greater than 3000**

**select first\_name,incentive\_amount from employee inner join incentive on employee\_id=employee\_ref\_id where incentive\_amount > 3000;**

* **Select 2nd Highest salary from employee table.**

**select max(salary) from employee where salary < (select max(salary) from employee );**

* **Select first\_name, incentive amount from employee and incentives table for all employees who got incentives using left join.**

**select employee.first\_name,incentive.incentive\_ammount from employee left join incentive i on i.employee\_ref\_id = e.employee\_id;**

* **Create View OF Employee table in which store first name ,last name and salary only.**

**create view Emp\_detail as**

**select first\_name,last\_name,slaary from employee;**

* **Create Procedure to find out department wise highest salary.**

**CREATE PROCEDURE max\_salary ()**

**BEGIN**

**select department, max(salary) from employee group by department**

**END**

* **Create After Insert trigger on Employee table which insert records in view table.**

**delimiter $$**

**create**

**trigger update after insert**

**on stundents**

**for each row begin**

**insert into new\_table\_name (id,name,email) values (old.id,old.name,old.email);**

**end$$**

**delimiter;**

* **All orders for more than $1000.**

**select \*from table\_name where amount>$1000;**

* **Names and cities of all salespeople in London with commission above 0.10.**

**select sname,city from table\_name where comm >0.10 and city='london' ;**

* **All salespeople either in Barcelona or in London.**

**select \*from salesperson where city in ('barcelona','lonodn');**

* **All salespeople with commission between 0.10 and 0.12. (Boundary values shouldbe excluded).**

**select \*from salesperson where comm between o.12 and 0.12;**

* **All customers excluding those with rating <= 100 unless they are located in Rome.**

**select \*from customer where rating <=100 or city ='rome';**

* **All orders except those with 0 or NULL value in amt field.**

**select \*from orders where amt!=0 or ant = not null;**

* **Count the number of salespeople currently listing orders in the order table.**

**select count(distinct sno) from orders;**

* **Largest order taken by each salesperson, datewise.**

**select max(amt) from orders group by ODE,sno order by ODE, sno;**

* **Largest order taken by each salesperson with order value more than $3000. (ie same city).**

**select max(amt) from orders where amt >3000;**